

Teknisk slutrapport för
FLTP-Framtidens LeveransTågplaneProcess

Sara Gestrelus, Martin Aronsson
RISE SICS AB, Box 1263, 164 29 KISTA
Email: förnamn.efternamn@sics.se

28 februari 2017

SICS Technical Report T2017:02

Sammanfattning

Slutrapporten sammanfattar arbetet som gjorts i projektet *FLTP-Framtidens LeveransTågplaneProcess*. FLTP har utförts av RISE SICS AB på uppdrag av Trafikverket under åren 2014-2016. Syftet med projektet har varit att undersöka planeringsmetoder för att ta fram leveransåtagande i en tänkt framtida långtidsprocess när Successiv Planering är fullt genomfört. Projektet har tagit fram ett planeringskoncept där alla kördagars individuella trafiksituation beaktas, och matematiska modeller och heuristiker som kan stödja den föreslagna planeringsmetoden har utformats och testats. Rapporten presenterar övergripande de metoder, resultat och slutsatser som projektet levererat, och ger också sammanhang till projektets övriga publikationer. Slutrapporten presenterar även i stora drag de mål som projektet satte upp, samt till vilken grad dessa har uppfyllts.



Innehåll

1	Inledning	3
1.1	Kort projekthistoria	3
1.2	Rapportupplägg	4
2	Mål för den framtida långtidsprocessen	4
3	Skiss för framtida process och planeringsmetod	5
4	Planera för varje dag individuellt	8
4.1	Ömsesidigt uteslutande konflikter	9
4.1.1	Analys av antalet uteblivna konflikter för T14	9
4.2	Testfall: daglig planering på sträckan Skymossen-Mjölby	15
4.2.1	Resultat	15
5	Automatiskt tidtabelläggning	18
5.1	Matematisk optimeringsmodell	18
5.1.1	Data i modellen	19
5.1.2	Tågs basekvationer	19
5.1.3	Konfliktreglering	20
5.1.4	Målfunktion	20
5.2	Rullande planering	21
5.3	Heuristiska metoder	22
5.3.1	Successiv tilläggs-och fixerings-heuristik (STFH)	22
5.3.2	Iterativ geografisk pusslingsheuristik (IGPH)	23
5.3.3	Förbättringsheuristik (FH)	27
5.4	Testfall: Hallsbergs driftledningsområde	29
6	Sammanfattning	30

1 Inledning

Projektet FLTP (Framtidens LeveransTågplaneProcess) är ett treårigt projekt finansierat av Trafikverket och utfört av RISE SICS AB. Projektets syfte är att undersöka planeringsmetoder för att ta fram leveransåtagande (senare kallat avtalstider) i en tänkt framtida långtidsprocess när Successiv Planering är fullt genomfört.

Successiv Planering (SP) innebär att Trafikverket fastställer och utlovar de tidtabellsdetaljer som är viktiga för operatörer och entreprenörer att ta ställning till, istället för att som idag fastställa alla detaljer i tidtabellen. Ett exempel på en viktig tidtabellsdetalj är ett passagerartågs ankomst till en station med passagerarutbyte, eller passagerartågets avgång från samma station. Passagerartåget måste ankomma innan ankomsttiden och avgå efter avgångstiden. Däremot är det inte viktigt när passagerartåget passerar en station utan kommersiell verksamhet. Genom att endast fastställa viktiga tidtabellsdetaljer i så kallade leveransåtagande, eller avtalstider, öppnas möjligheter för att effektivisera kapacitetsutnyttjandet. Till exempel kan den nyvunna flexibiliteten användas i en driftoptimering som flyttar buffertid till de tidtabellspunkter där den behövs mest just den aktuella driftdagen, och i ad hoc-processen kan flexibiliteten användas för att få plats med nya tåg och banarbeten på ett bättre sätt. Även långtidsprocessen bör utnyttja den flexibilitet som SP innebär. Långtidsprocessen borde använda de möjligheter som SP ger för att skapa attraktiva leveransåtagande till sökande, men den ska även säkerställa att de leveransåtagande som erbjuds och fastställs ger bra förutsättningar för ad hoc-processen och den slutgiltiga driften.

Den här slutrapporten sammanfattar det arbete som utförts inom FLTP och ger också sammanhang till projektets övriga publikationer. Den presenterar också i stora drag de mål som projektet satte upp, samt hur dessa har uppfyllts.

1.1 Kort projekthistoria

Framtidens LeveransTågplaneProcess (FLTP) är det första forskningsprojektet som fokuserat specifikt på en planeringsmetod till långtidsprocessen i en Successiv Planerings-miljö. FLTP startade i början av 2014, och tog då fram en omvärldsanalys med fokus på supply chain management i övrig industri [3]. Projektet fortsatte sedan med att ta fram en problemspecifikation och göra initial modellering och implementering. I slutet av 2014 fanns en grov struktur för hur en framtida långtidsprocess skulle kunna se ut, och en matematisk modell som skulle kunna ingå i det beslutstöd som krävs i denna process. Den föreslagna metoden går ut på att varje dag planeras individuellt. Ett tåg kan således få olika planerade tåglägen olika dagar, och vi kallar dessa tåglägen för "tåglägeskog". Tåglägeskogen analyseras för att bestämma vilka leveransåtagande som Trafikverket bör erbjuda operatörer och entreprenörer. De första stadierna av arbetet presenterades på operationsanalyskonferensen IFORS i Barcelona, samt på nationella transportforskningskonferensen i Norrköping och slutresultatet publicerades i Gestrelus et al. [4].

Under projektets andra år låg fokus på resultatpresentation samt validering och utveckling av modellen. Till exempel gjordes ett omtag på geografimodelleringen för att kunna hantera tåg på högerspår. Projektet presenterade resultat

på Transportforum, RailTokyo, SICS Open House, KAJT dagarna samt på ITS-EASY Annual Symposium.

Under projektets sista år, 2016, låg fokus på att hitta metoder för att automatiskt generera tidtabeller för större geografiska problemområden. Två heuristiker och en förbättringsheuristik designades och implementerades. Dessa heuristiker presenterades på KAJT-dagarna och en publikation med den bästa heuristiken har blivit accepterad i det första steget till European Working Group On Transportation 2017 i Budapest [5].

1.2 Rapportupplägg

Rapporten inleds med en beskrivning av de mål som sattes upp i början av projektet (avsnitt 2). I avsnitt 3 introduceras den process-skiss som togs fram under projektets första år, inklusive den ansats till ny planeringsmetod som projektet jobbat med. Ansatsen går ut på att varje dag planeras individuellt med hjälp av automatiserad tidtabellläggning. Avsnitt 4 är en närmare analys av hur daglig planering kan leda till effektivare leveransåtagande. Konceptet *ömsesidigt uteslutande konflikter* introduceras och vi ger en grov beskrivning av hur dessa kan påverka leveransåtagande i dagens process och i den skissade långtidsprocessen. Resultaten från en grov analys av ömsesidigt uteslutande konflikter i T14 presenteras. Avsnitt 4 avslutas med att den föreslagna metoden med daglig planering testas i en fallstudie på sträckan Skymossen-Mjölby.

Eftersom den skissade processen kräver automatiserad tidtabellläggning har en stor del av projektets arbete gått ut på att utforma och implementera optimerande och heuristiska metoder för detta ändamål. Detta arbete beskrivs i avsnitt 5. Avsnitt 5 inleds med att en optimeringsmodell och rullande planering presenteras. Det var dessa tekniker som användes i fallstudien Skymossen-Mjölby. Eftersom optimeringsmodellen inte kan hantera stora geografiska områden utvecklades även två heuristiker för tidtabellsgenerering samt en förbättringsheuristik. Dessa presenteras i stycke 5.3, inklusive resultat från den testkörning som gjordes på hela Hallsbergs driftledningsområde.

Rapporten avslutas med en diskussion av vilka av de identifierade målen som har uppnåtts, samt vilka utmaningar som kvarstår (avsnitt 6).

2 Mål för den framtida långtidsprocessen

I dagens långtidsprocess planeras alla tåglägen på sekundnivå för en dag med alla tåg som går under året. Vi kallar denna dag "årstypdag". Alla konflikter som sker fler än ett fåtal gånger under året regleras. Årstypdagsplanen är mer detaljerad än vad som verkar rimligt men beaktar ändå inte att trafiksituationen skiljer sig under året (olika tåg körs under olika perioder). De tåglägen som fastställs förväntas dessutom förbli exakt desamma under hela året, och förväntas sättas i drift som de är planerade. I SP fastslås leveransåtagande istället för hela tåglägen. På så sätt bevaras friheten att lägga om och optimera tåglägen för varje dags unika trafiksituation. Det gör också att det är lättare att utnyttja den kapacitet som finns tillgänglig till t.ex. ad hoc-tåglägen. Detta är sant oavsett hur leveransåtagandena tas fram, d.v.s. det är nyttor som kan nås även om den enda förändringen i dagens långtidsprocess är att leveransåtagande fastställs istället för hela tåglägen.

Vi tror dock att det går att ta fram leveransåtagande som på ett bättre sätt utnyttjar SPs flexibilitet än vad dagens process tillåter. Tanken är att utveckla en planeringsmetod som på ett bättre sätt:

1. Beaktar att trafiksituationen skiljer sig olika dagar, och inte binder upp kapacitet i onödan.
2. Sparar kapacitet till ad hoc-tåg.

Vidare tror vi att den framtida långtidsprocessen kan vara annorlunda än dagens process. Vi tror Trafikverket i framtiden kommer vilja erbjuda leveransåtagande med olika kvaliteter beroende på de behov som den sökande har. Vidare kan processen innehålla någon sorts förhandlingar eller auktioner. Tanken är att den framtida långtidsprocessen ska kunna vara mer iterativ än dagens, och låta sökande och Trafikverket testa olika förutsättningar och upplägg för att i slutändan nå en lösning som på ett bättre sätt uppfyller både sökandes och Trafikverkets behov. Därför tror vi att det är viktigt att de prototyper för stödverktyg som tas fram kan generera tåglägen med olika kvaliteter, och har exekveringstider som är korta nog att tillåta en iterativ process. Vi antar också att kravet på konfliktfrihet kvarstår, dvs. en konfliktfri tidtabell som uppfyller alla leveransåtagande måste gå att finna för varje driftdag.

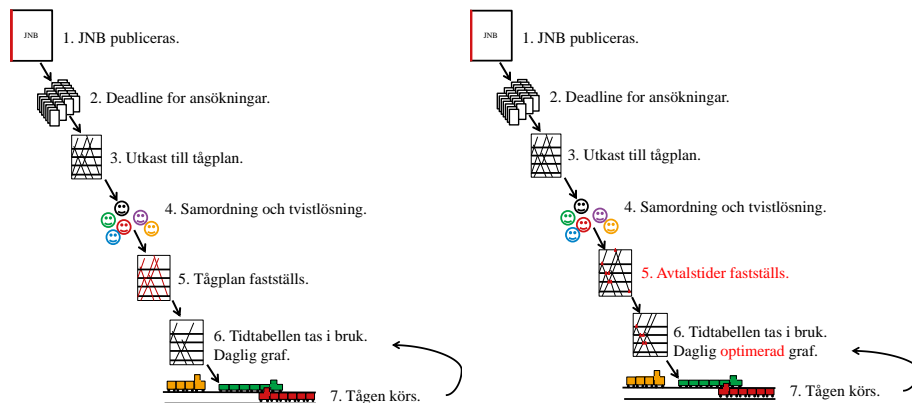
Baserat på dessa resonemang har fyra mål för en bra planeringsmetod för långtiden identifierats. Planeringsmetoden ska:

1. Vara snabb.
2. Ta fram leveransåtagande som uppfyller olika behov för olika sökande.
3. Ta fram leveransåtagande som ger ett lösbart tidtabellproblem varje driftdag.
4. Ta fram leveransåtagande som använder spårkapaciteten effektivt.

3 Skiss för framtida process och planeringsmetod

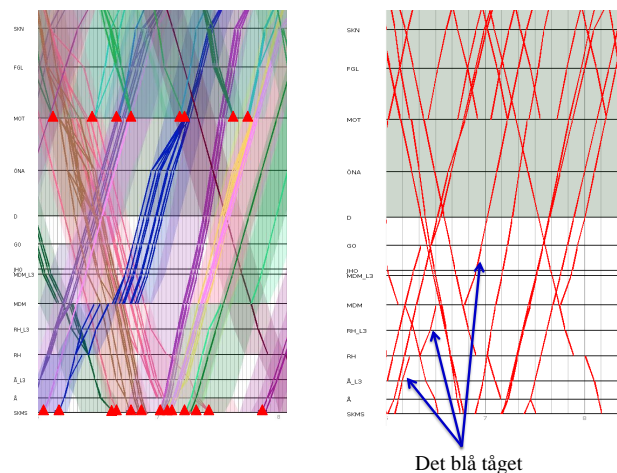
Den stora skillnaden mellan dagens process och en framtida process med Successiv Planering är att leveransåtagande istället för alla produktionsdetaljer i tåglägen fastställs. Figur 1 visar denna skillnad givet att långtidsprocessen förblir den samma. Att fastställa leveransåtaganden istället för alla produktionsdetaljer innebär att tåg kan gå olika olika dagar. Detta visas i Figur 2. Denna flexibilitet kommer i första hand utnyttjas i ad hoc-processen och i den robusthetsoptimering som ska genomföras innan daglig graf överlämnas, men den borde även beaktas i långtidsprocessen.

Figur 3 visar den processskiss som FLTP tog fram. Planeringsmetoden börjar efter att auktoriserade sökande skickar in sina ansökningar (Steg 1). Utifrån ansökningarna genereras ett antal planeringsproblem (Steg 2). De olika planeringsproblemen motsvarar olika tidsperioder. Tidsperioderna kan t.ex. vara sommar och vinter, kvartal, eller veckodagar. I det mest extrema fallet är varje enskild dag ett eget planeringsproblem. Målet med att dela upp tidtabellsåret i flera perioder är att beakta de kördagsvariationer som finns. Vilka perioder som



(a) I dagens process fastställs alla produktionsdetaljer i tåglägena. (b) I morgondagens process fastställs leveransåtagande, t.ex. avtalsider.

Figur 1: Skillnaden mellan dagens och morgondagens process om långtidsprocessen inte förändras förutom att endast leveransåtagande fastställs.



Figur 2: Om man fastställer leveransåtagande (röda trianglar) istället för hela tåglägen kan det blå tåget, och alla andra tåg, gå olika olika dagar (vänster bild). Detta ger en flexibilitet som kan användas i både långtidsprocessen, ad hoc-processen och vid driftoptimering. Om däremot hela tåglägen fastställs måste alla tåglägen se likadana ut varje dag (höger bild), och ingen flexibilitet finns.

väljs ut har således stor betydelse för hur pass attraktiva leveransåtaganden som kan erbjudas, och kördagarna i en period bör vara så lika varandra som möjligt. FLTP har jobbat med den mest extrema varianten, där varje kördag planeras individuellt och inga kördagar klumpas ihop som typdagar. Det är denna extrema variant som kommer presenteras och undersökas i rapporten.

Efter att tidtabellsåret delats in i tidsperioder (i vårt fall dagar) genereras en tidtabell för varje planeringsproblem (Steg 3), och den tåglägeskog som ett

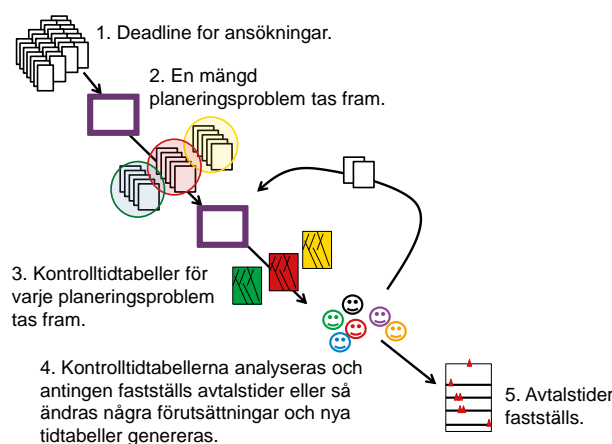
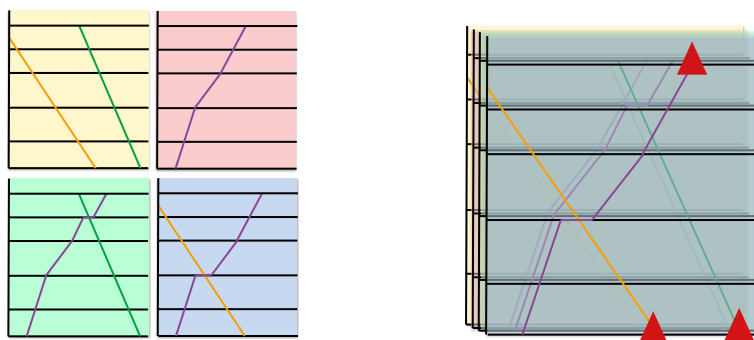


Figure 3: Skiss av föreslagen planeringsmetod för långtiden.

visst tåg har analyseras för att avgöra vilka leveransåtagande och avtalstider som ska erbjudas (Steg 4). En möjlig analysmetod är att alltid välja den sämsta tiden, worst-case tiden (se Figur 4). Eftersom det alltid finns en tidtabell som uppfyller denna worst-case situation så garanteras att en konfliktfri tidtabell kan genereras för alla kördagar. En annan metod är att börja använda konceptet uppfyllnadsgrad. *Uppfyllnadsgrad* är hur pass många dagar av det totala antalet kördagar som Trafikverket lovar att tåget uppfyller sina avtalstider i den dagliga grafen. T.ex. kanske en operatör hellre vill ha ett leveransåtagande med en kort körtid och en 90% uppfyllnadsgrad än ett leveransåtagande med en lång körtid och en 100% uppfyllnadsgrad. Samma koncept med uppfyllnadsgrad kan även användas för att t.ex. göra rum för sena banarbeten eller prioritera tåg vid snöfall. Notera att om leveransåtaganden fastställs med en uppfyllnadsgrad som är lägre än 100% så garanterar planeringsmetoden inte att det alltid går att hitta en konfliktfri tidtabell som uppfyller alla avtalstider eftersom vi räknat med att ett tåg inte kommer uppnå sina avtalstider under en viss andel dagar. Dock kommer leveransåtagandet, som specificerar inte bara avtalstiderna utan också uppfyllnadsgraden, alltid kunna upprätthållas i daglig graf på årsbasis. Ytterligare ett verktyg som finns tillgängligt är att dela upp ett tåg i varianter. Om tågläggesskogen har stor spridning, eller är tydligt uppdelad, så kan detta indikera att tåget borde delas upp i varianter.

Det är viktigt att inse att de tidtabeller som tas fram i Steg 3 i processkissen inte är produktionstidtabeller utan *kontrolltidtabeller*. Kontrolltidtabellerna är argument för att ett leveransåtagande kan erbjudas. De slutgiltiga produktionstidtabellerna kommer innehålla fler tåg som tillkommit i ad hoc-processen, och kommer också ha genomgått den robusthetsoptimering som görs innan den dagliga grafen överlämnas.

Loopen i skissen som tar processen tillbaka från Steg 4 till Steg 3 representerar att långtidsprocessen kan komma att vara iterativ. Om Steg 3 och 4 är tillräckligt snabba så kan de erbjudna leveransåtagandena diskuteras med operatörer och entreprenörer, och nya upplägg testas. Poängen är att genom en iterativ process möjliggöra ett bättre samspel mellan operatörer, entreprenörer och Trafikverket.



(a) Olika dagar har olika tidtabeller. (b) Genom att analysera spridningen av ankomsttider och avgångtider kan avtalstider genereras. Till exempel kan de sämsta, worst-case, tiderna utlovas (markerade med röd triangel).

Figur 4: Skogen av tåglägen analyseras för att bestämma avtalstider. Denna bild representerar steg 4 i bild 3.

Planeringsmetoden som beskrivits ovan kan bli svår att hantera om tåglägeskogen blir allt för spridd, och vissa udda aspekter kan uppstå, så som att worst-case ankomsttiden (dvs. den senaste) är senare än worst-case avgångstiden (dvs. den tidigaste). Detta är något man bör beakta och kanske till och med hantera i optimeringsmodellen och beslutsstödet i en verklig implementation.

4 Planera för varje dag individuellt

Det finns två aspekter som avgör huruvida det är bra att planera för varje dag individuellt i längtiden, dels hur pass stor potentialen är, dvs. hur pass mycket bättre transport- och banarbetsleveransåtagande som kan erbjudas med dagligt anpassad planering, och dels om själva ansatsen är möjlig att genomföra eller inte.

I avsnitt 4.1 presenteras *ömsesidigt uteslutande konflikter*. Ömsesidigt uteslutande konflikter är den förutsättning som krävs för att planering med en årstypdag ska ge mer mindre effektiva leveransåtagande än om varje dag hanteras individuellt. Dessa kan således analyseras för att få en känsla för om det finns någon potential för daglig planering eller inte. Den generella idén är att om det finns många och långa ömsesidigt uteslutande konflikter som påverkar ett tåg så är det sannolikt att detta tåg kan få bättre leveransåtagande om kördagsvariationer beaktas i långtidsprocessen. I FLTP genomfördes bara en mycket grov analys av ömsesidigt uteslutande konflikter, och resultaten presenteras i avsnitt 4.1. I avsnitt 4.2 beskrivs sedan det första försök som gjordes vad gäller daglig planering för att generera leveransåtagande. Resultaten som beskrivs i avsnitt 4.2 är publicerade i [4].

4.1 Ömsesidigt uteslutande konflikter

Som påpekats tidigare så kör inte alla tåg alla dagar i en tågplaneperiod. Detta innebär att den trafik som möter ett specifikt tåg A kommer variera, och således kommer den konflikthantering som krävs för att tåg A ska ha ett konfliktfritt tågläge också variera. Genom att utnyttja den kapacitet som faktiskt finns de olika dagarna kan tåg A få bättre leveransåtagande.

Antag att tåg A är ett lågt prioriterat tåg som kör på en enkelspårsbana alla dagar i veckan. Operatören önskar att tåg A ska ha en så kort körtid som möjligt. Antag vidare att tåg A möter ett tåg B som bara kör på måndagar och ett tåg C som bara kör på onsdagar (se Figur 5a). Eftersom tåg A är lågt prioriterat kommer det få stanna för båda mötena. Men all den konfliktregleringstid som planeras för tåg A kommer aldrig behövas eftersom mötet ($A \times B$) och mötet ($A \times C$) aldrig sker samma dag. Konflikterna är fullkomligt ömsesidigt uteslutande. Vidare finns det partiellt ömsesidigt uteslutande konflikter. En sådan situation visas i Figur 5b. Tåg A går precis som tidigare varje dag. Tåg B går däremot måndag och tisdag, tåg D tisdag och onsdag, och tåg C onsdag och måndag. Detta innebär att alla konflikter är partiellt ömsesidigt uteslutande.

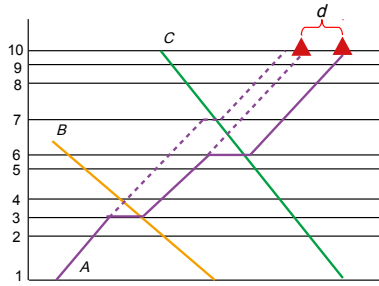
I exemplet ovan med tåg A så var alltid tåg A själv inblandat i de ömsesidigt uteslutande konflikterna. Men tåg A kan påverkas även av konflikter som det inte är inblandat i. Figur 5c är ett exempel på detta. Konflikterna ($B \times D$) och ($A \times C$) är fullkomligt ömsesidigt uteslutande, och även om tåg A inte är direkt inblandat i konflikten mellan tåg B och D så kommer spridningseffekten leda till att daglig planering i långtiden genererar ett leveransåtagande med kortare körtid än dagens process.

Huruvida fullkomligt eller partiellt uteslutande konflikter orsakar längre körtid än nödvändigt beror på tidtabellens helhet. Figur 5d visar en situation där körtiden inte kan minskas även om daglig planering används i en situation med fullkomligt ömsesidigt uteslutande konflikter.

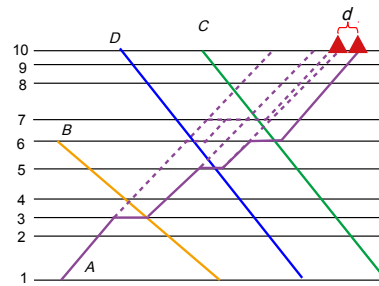
Som påpekats tidigare så är den bakomliggande tanken att om ett tåg påverkas av många och långa ömsesidigt uteslutande konflikter är det sannolikt att detta tåg kan få bättre leveransåtagande om kördagsvariationer beaktas i långtidsprocessen. Eftersom det är tidtabellens helhet som avgör vilka uteslutande konflikter som påverkar ett tågs leveransåtagande så är effekten av uteslutande möten svår att fånga. I nästa avsnitt räknar vi därför antalet uteblivna konflikter, som är en förutsättning för ömsesidigt uteslutande möten, snarare än ömsesidigt uteslutande möten.

4.1.1 Analys av antalet uteblivna konflikter för T14

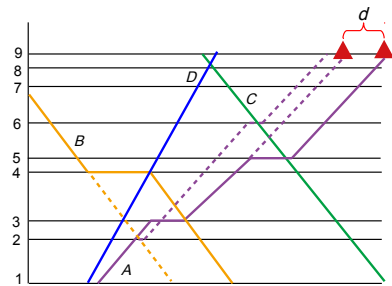
Eftersom det är svårt att avgöra effekten av ömsesidigt uteslutande konflikter analyseras istället antalet uteblivna konflikter i T14 för två passagerartåg och två godståg. Till grund för analysen ligger så kallade *planerade konfliktstopp*. Ett planerat konfliktstopp är ett stopp som är inplanerat **endast** av konfliktregleringsskäl. Ett stopp med ett kommersiellt syfte räknas således inte som ett planerat konfliktstopp, även om t.ex. en omkörning görs under det kommersiella stoppet. Ett konfliktstopp kan reglera flera konflikter, och vi har därför även räknat antalet konflikter som regleras med konfliktstopp. Den tid som tåget står stilla vid varje konfliktstopp registreras också. Vi kallar denna tid för *stilleståndstid*, och använder den för att uppskatta stoppets inverkan på tåget. Notera att detta



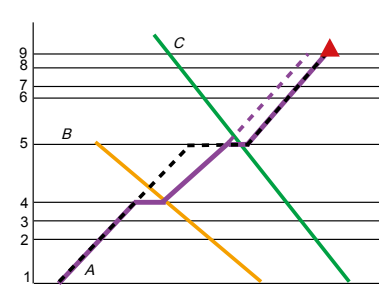
(a) Tåg A möter tåg B och tåg C, men tåg B går på måndagar och tåg C på onsdagar. Konflikterna $(A \times B)$ och $(A \times C)$ är således ömsesidigt uteslutande och tåg A kommer aldrig möta både tåg B och C.



(b) Tåg B går måndag och tisdag, tåg D tisdag och onsdag, och tåg C onsdag och måndag. Detta innebär att alla konflikter är partiellt ömsesidigt uteslutande. Eftersom alla tre konflikter aldrig sker samma dag så kan tåg As körtid minskas i detta exempel.



(c) Konflikterna $(B \times D)$ och $(A \times C)$ är fullkomligt ömsesidigt uteslutande. Även om tåg A inte är direkt inblandat i konflikten mellan tåg B och D så leder spridningseffekten till att daglig planering genererar ett leveransåtagande med kortare körtid för tåg A.



(d) Konflikterna $(A \times B)$ och $(A \times C)$ är fullkomligt ömsesidigt uteslutande. Eftersom mötet med tåg C kommer tas på station 5 oavsett om A först möter B (heldragen linje) eller inte (svart streckad linje) så kan körtiden i leveransåtagandet inte minskas. Körtiden kan minskas de dagar som tåg A bara möter tåg B.

Figur 5: Olika exempel på hur ömsesidigt uteslutande konflikter påverkar ett tåg A som kör alla dagar. Den heldragna linjen visar det tågläge som reglerar alla konflikter, medan de streckade visar den konfliktreglering som faktiskt behövs varje individuell kördag. Bokstaven d visar hur pass mycket tåg As körtid kan minskas i exemplet. Om tåg A ska köra längre än exemplet visar kan det finnas andra konflikter senare som förhindrar denna körtidsminskning.

är förenklingar. I en riktig uträkning bör tid för inbromsning och acceleration, långsammare körning på linje och extra tid på mötespunkter med kommersiell aktivitet också inkluderas. Ytterligare en förenkling vi gjort är att om ett konfliktstopp reglerar minst en konflikt en given kördag, då antar vi att all planerad stilleståndstid behövs för konfliktstoppet den dagen.

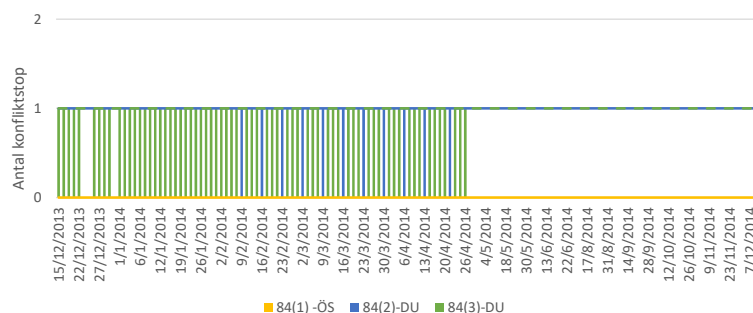
I Figur 6-8 visas antalet konfliktstopp, antalet konflikter som regleras med konfliktstopp och summan av stilleståndstid för alla konfliktstopp med minst en konflikt för varje kördag för ett passagerartåg och två godståg. Det andra persontåget, som körde mellan Stockholm och Hudiksvall/Umeå, hade aldrig några planerade konfliktstopp, och finns således inte representerat i någon graf.

Figur 6 visar konfliktstoppsdata för ett passagerartåg mellan Stockholm och Östersund/Duved. Passagerartåget har bara ett konfliktstopp (i två av sina varianter) i den årliga tågplanen och kommer således behöva antingen all eller ingen stilleståndstid. Konfliktstoppet reglerar dessutom bara en konflikt, vilket gör att grafen med antalet konfliktstopp och grafen med antalet konflikter som reglerats vid konfliktstopp är samma. Tåg som har få eller inga konfliktstopp kommer antagligen inte få märkbart mycket bättre leveransåtagande i en långtidsprocess som beaktar kördagsvariationer. Det skulle i så fall komma sig av att dessa tåg påverkas negativt av andra konfliktregleringsmetoder i dagens process, så som t.ex. minskad hastighet på linjen.

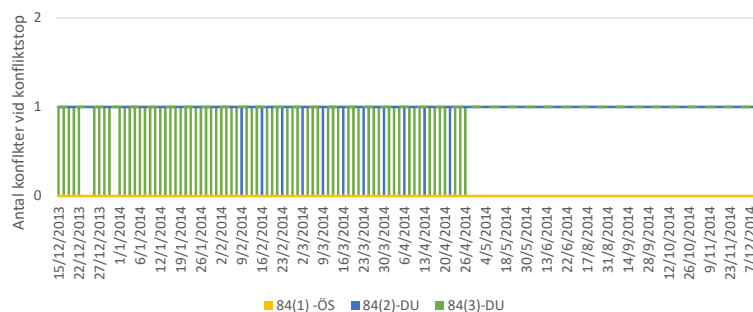
Godstågens mötesbild är oftast mer varierande. Det totala antalet konfliktstopp är 12 och 5 för varianterna av tåget mellan Luleå och Domnarvet, och 3 och 4 för varianterna av tåget mellan Sävenäs och Sundsvall. För båda tågen så används samma konfliktstopp för att reglera flera konflikter. Det totala antalet konflikter som regleras med konfliktstopp är 27 och 14 för tåget mellan Luleå och Domnarvet och 3,4 och 5 för tåget mellan tåget mellan Sävenäs och Sundsvall.

I Figur 7 syns konfliktstoppsdata för godståget mellan Luleå och Domnarvet. Ofta blir många av de planerade konfliktstoppen av, men aldrig alla och all den stilleståndstid som planerats in i tågläget behövs därför aldrig för konfliktreglering. Det här är ett tåg som kanske kan få bättre leveransåtaganden om daglig planering används i långtidsprocessen.

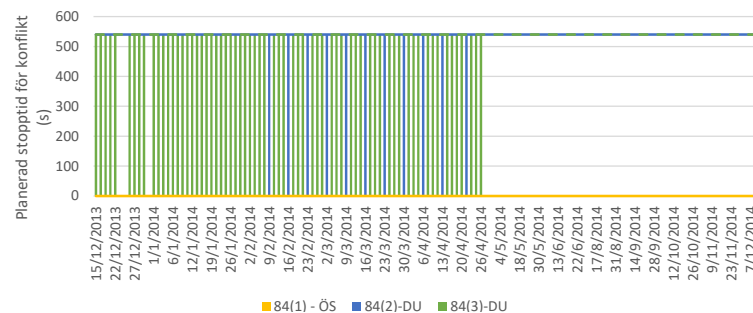
I Figur 8 syns konfliktstoppsdata för godståget mellan Sävenäs och Sundsvall. För två av tågets varianter (den blå och den gula) finns det kördagar då alla konfliktstopp blir av, men för den gröna varianten är det många kördagar där den planerade stilleståndstiden är mycket större än den stilleståndstid som finns i de konfliktstopp som blir av varje kördag. Den gröna varianten av tåget, 4584(3), kan med största sannolikhet få bättre leveransåtaganden om daglig planering används i långtidsprocessen.



(a) Staplarna visar antalet planerade konfliktstopp som reglerar minst en konflikt under kördagen. Linjen visar antalet planerade konfliktstopp i det fastställda tågläget.

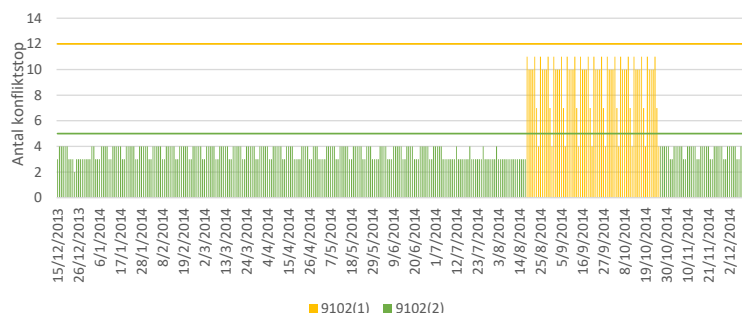


(b) Staplarna visar antalet konflikter som regleras med planerade konfliktstopp under kördagen. Linjen visar antalet konflikter som regleras med planerade konfliktstopp i det fastställda tågläget.

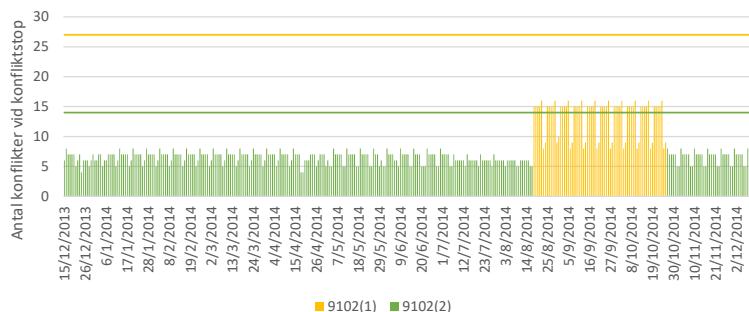


(c) Staplarna visar summan av stilleståndstid för planerade konfliktstopp med minst en konflikt under kördagen. Linjen visar stilleståndstiden för planerade konfliktstopp i det fastställda tågläget.

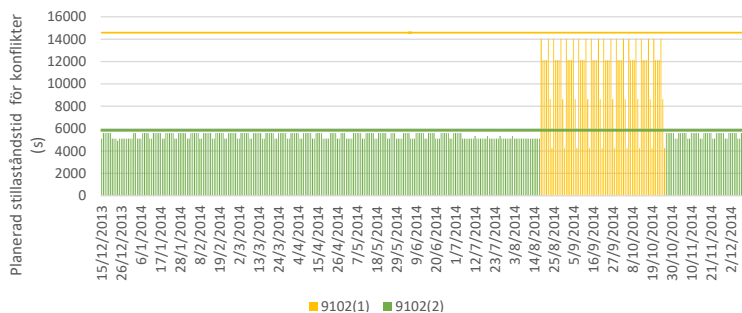
Figur 6: Passagerartåg Stockholm-Östersund/Duved.



(a) Staplarna visar antalet planerade konfliktstopp som reglerar minst en konflikt under kördagen. Linjen visar antalet planerade konfliktstopp i det fastställda tågläget.

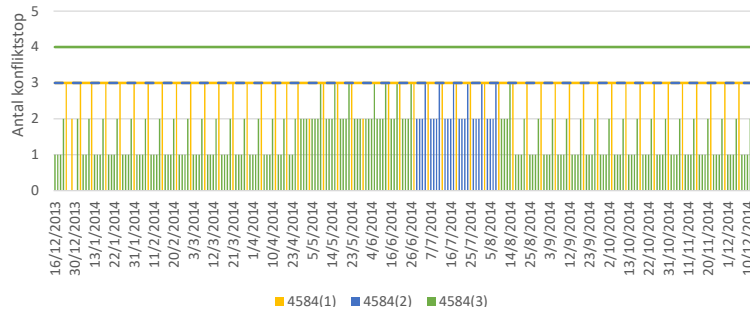


(b) Staplarna visar antalet konflikter som regleras med planerade konfliktstopp under kördagen. Linjen visar antalet konflikter som regleras med planerade konfliktstopp i det fastställda tågläget.

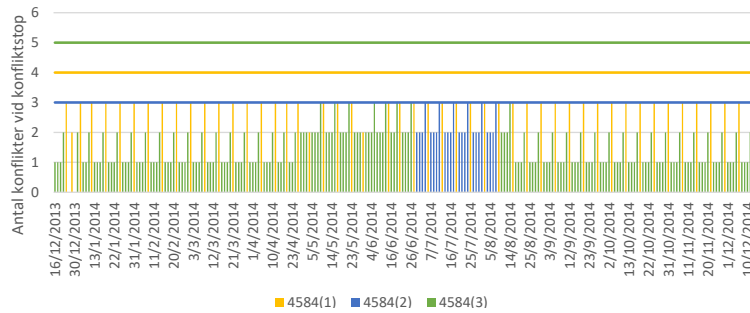


(c) Staplarna visar summan av stilleståndstid för planerade konfliktstopp med minst en konflikt under kördagen. Linjen visar stilleståndstiden för planerade konfliktstopp i det fastställda tågläget.

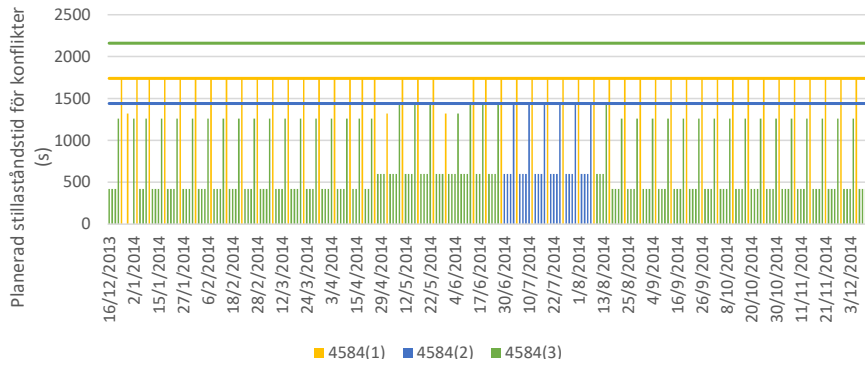
Figur 7: Godståg Luleå-Domnarvet.



(a) Staplarna visar antalet planerade konfliktstopp som reglerar minst en konflikt under kördagen. Linjen visar antalet planerade konfliktstopp i det fastställda tågsläget.



(b) Staplarna visar antalet konflikter som regleras med planerade konfliktstopp under kördagen. Linjen visar antalet konflikter som regleras med planerade konfliktstopp i det fastställda tågsläget.



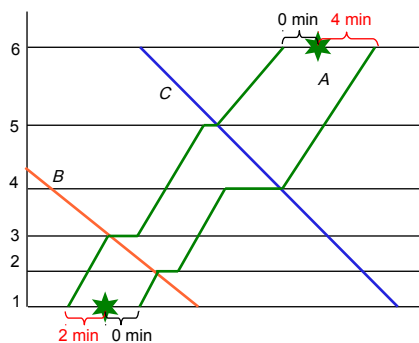
(c) Staplarna visar summan av stillaståndstid för planerade konfliktstopp med minst en konflikt under kördagen. Linjen visar stillaståndstiden för planerade konfliktstopp i det fastställda tågsläget.

Figur 8: Godståg Sävenäs-Sundsvall.

4.2 Testfall: daglig planering på sträckan Skymossen-Mjölby

För att testa om en långtidsprocess där varje dag planeras individuellt med hjälp av optimering är en framkomlig väg gjordes ett försök på sträckan Skymossen-Mjölby. Testfallet baseras på T14 vid fastställelse. Sträckan Skymossen-Mjölby valdes eftersom den består av både enkelspår och dubbelspår och har blandad trafik. På sträckan gick 214 tåg. 66 av dessa tåg valdes ut att vara tåg där operatören ansökte om att få en så kort körtid som möjligt medan övriga 148 tåg antogs vara ansökta med önskemål om avtalstider på viktiga punkter. För de tåg där de exakta önskade avtalstiderna var viktigast minimerades *feltiden*. Feltid är hur pass mycket tidigare avgång eller senare ankomst som kan erbjudas i leveransåtagandet jämfört med den önskade tiden (se Figur 9). Alla tåg var dessutom tvingade att ligga inom ± 15 minuter från det fastställda tågläget. Eftersom vi inte hade tillgång till ansökningarna så genererades ansökningar genom att tåglägena flyttades slumpmässigt inom en ± 10 minuters domän.

En tidtabell för tidsperioden 1 januari till 12 december 2014 togs fram med hjälp av optimering och rullande planering (se kapitel 5.1 och 5.2). Skogen av tåglägen för varje tåg analyserades och worst-case situationen (antingen ankomst/avgångs-tid eller körtid) valdes som leveransåtagande.



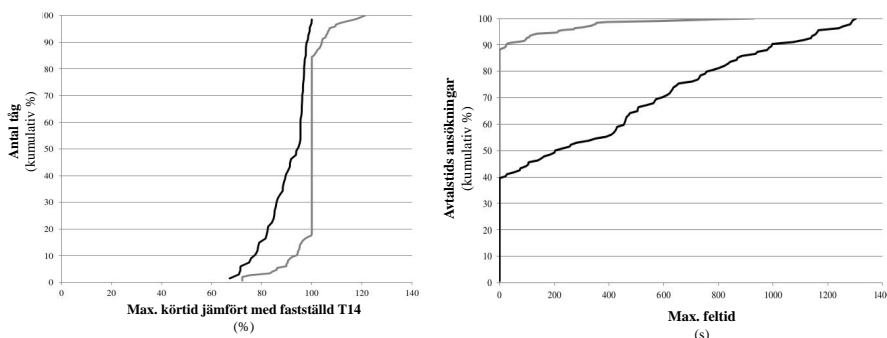
Figur 9: Bilden visar feltid för två varianter av det gröna tåget. Stjärnorna på station 1 och 6 representerar de önskade tiderna. Om det gröna tåget avgår för tidigt från station 1 så räknas det som feltid (röd tid på 2 minuter), men om det avgår senare än den önskade tiden räknas feltiden som 0 (svart tid på 0 minuter). Likaså är det dåligt om det gröna tåget ankommer till station 6 senare än önskat (röd tid på 4 minuter), men det anses inte vara några problem om tåget ankommer för tidigt (svart tid på 0 minuter).

4.2.1 Resultat

Den första frågan att besvara är om den föreslagna metoden kan ta fram leveransåtagande som svarar väl mot sökandes önskningar, dvs. om de operatörer som önskat en kort körtid i större utsträckning fick kortare körtid, och om de operatörer som önskat exakta avtalstider i större utsträckning fick dessa. Denna fråga knyter an till mål 2 i mållistan i avsnitt 2.

Figur 10 visar att den kvalitet som en operatör efterfrågat också återfinns i de leveransåtagande som erbjuds operatören. I Figur 10a syns körtiden i leveransåtagandet som daglig planering genererat som procent av körtiden i den

fastställda tågplanen. Körtiden i leveransåtagandet är kortare för alla tåg där körtid är den viktigaste kvaliteten, och för ca 15% är förbättringen mer än 20%. För tåg där överensstämmelse med önskade avtalstider är viktigast syns inte samma resultat. Vissa av dessa tåg får en kortare körtid, och andra inte. Figur 10b visar istället den feltid som leveransåtagandena har på alla avtalspunkter. För ett tåg där de önskade avtalstiderna är viktiga bör feltiden vara liten, vilket den också är. Däremot är feltiden inte liten för tåg där kort körtid är viktigast. Dessa resultat visar att metoden kan ta fram leveransåtagande som uppfyller olika önskingar för olika sökande.



(a) Operatörer som önskat korta körtider får kortare körtider. Jämförelse tider på avtalspunkter får detta med körtider i fastställd T14. (b) Operatörer som önskat specifika avtalstider får kortare körtider. Jämförelse tider på avtalspunkter får detta med körtider i fastställd T14.

Figur 10: Olika kvaliteter på leveransåtagande beroende på vad operatören önskat. Det svarta strecket visar tåg vars körtid minimerats, medan den grå linjen visar tåg vars feltid mot ansökan minimerats.

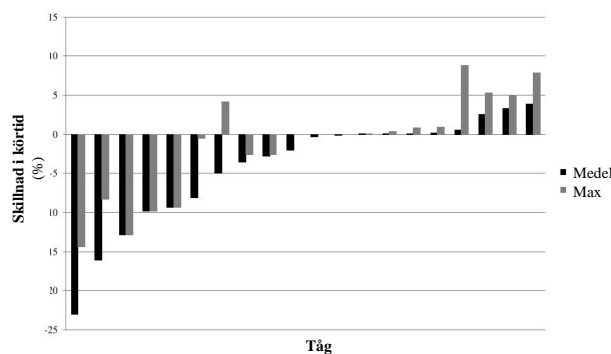
För att undersöka om kapacitetsutnyttjandet blir mer effektivt med daglig planering användes optimeringsmodellen för att generera en bastidtabell som representerar dagens långtidsprocess. Bastidtabellen togs fram genom att optimera fram en tidtabell för en typdag med alla tåg i. Alla konflikter som kunde uppstå någon gång under året reglerades. Vi definierar *effektiv kapacitetstilldelning* som kapacitetstilldelning som genererar leveransåtagande som i större utsträckning tillgodoser operatörernas behov, och jämför sedan de leveransåtagande som worst-case analysen genererat med de leveransåtagande som tiderna i bastidtabellen ger. Skillnaden mellan de två metoderna syns i Figur 11, där en negativ stapel innebär att leveransåtagandet som genererats med daglig planering är bättre än det som bastidtabellen kan ge. Både worst-case tiden (grå) och medelvärdestiden (svart) för tåglägeskogen visas i grafen.

För 46 av tågen där kort körtid var den viktigaste kvaliteten påverkades inte körtiden av planeringsmetoden. För övriga tåg visas skillnaden i Figur 11a. För 8 tåg minskade worst-case körtiden (grå), för 3 tåg var den samma, och för 9 tåg ökade den. Dock ökade körtiden generellt mindre än vad den minskade. Tittar man istället på medelvärdestiden (svart) så minskade denna för 12 tåg och ökade för 8. I de fall där medelvärdestiden och worst-case tiden är väldigt olika kan man tänka sig att worst-case tiden orsakas av att trafikläget under ett fåtal kördagar är sådant att tåget missgynnas, och då får en ovanligt lång körtid.

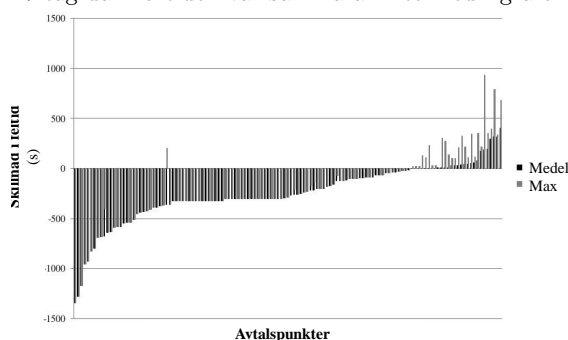
Att ett fåtal kördagar kan ha en så pass negativ effekt på leveransåtagandet är något som bör hanteras i en verklig process. T.ex. kan en kvadratisk istället för linjär målfunktion användas, eller så kan något omkringliggande ramverk försöka styra optimeringen så att tåg inte får exceptionellt dåliga tåglägen ett fåtal gångdagar. Ett annat sätt att arbeta med problemet är dela upp tåget i flera varianter eller att använda sig av leveransåtagande med uppfyllnadsgrad, så som diskuterats i avsnitt 3.

Figur 11b visar hur feltiderna påverkas av de olika planeringsmetoderna. Vad gäller feltider är det tydligt att daglig planering generellt ger bättre leveransåtagande än bastidtabellen. Men återigen återfinns problemet med att några tåg får en mycket sämre worst-case tid än medeltiden.

Att majoriteten av tågen får bättre eller opåverkade leveransåtagande med daglig planering jämfört med bastidtabellsplanering stärker vår hypotes att den föreslagna planeringsmetoden kan tilldela kapacitet på ett effektivt sätt. Dock kvarstår vissa viktiga frågor, så som t.ex. hur man ska hantera tåg vars tåglägeskög har ett fåtal exceptionellt dåliga tåglägen.



(a) Skillnaden mellan körtiden från bastidtabellen och körtiden i leveransåtagandet från daglig planering. De 47 tåg där körtiden var samma är inte med i grafen.



(b) Skillnaden i feltid mellan bastidtabellen och leveransåtagandet från daglig planering.

Figur 11: Att planera för varje dag individuellt leder till ett mer effektivt kapacitetsutnyttjande. Worst-case tiden visas i grått och medeltiden i svart.

Vad gäller exekveringstiderna i datorn så löstes 90% av de 695 problemin-

stanserna inom 280 sekunder. Dock fanns det två instanser vars optimala lösning inte kunde hittas inom exekveringstiden på 30 minuter. För dessa två instanser var optimalitetsgapet 0.01% och 0.24% när tidsgränsen stängde ner optimeringen (se avsnitt 5.1 för en beskrivning av optimalitetsgap). Dessa exekveringstider är acceptabla, men när geografins storlek ökar blir optimeringens exekveringstider snabbt ohanterligt långa.

5 Automatiskt tidtabellläggning

För att ta fram en kontrolltidtabell för varje dag krävs det någon form av automatisk planering. Det lägsta kravet är att den automatiska planeringsmetoden tar fram en giltig tidtabell. En giltig tidtabell är en konfliktfri tidtabell som respekterar alla de fysiska krav som infrastrukturen, banarbetena och fordonen ger. Olika giltiga tidtabeller har olika egenskaper, och dessa egenskaper kan vara mer eller mindre önskvärda. Till exempel kan tidtabeller kan ha olika robusthet och uppfylla kundernas önskningar olika väl. En målfunktion används för att avgöra hur pass bra en tidtabell är. Om målfunktionen t.ex. är summan av alla tågens körtider så kommer en tidtabell med korta körtider utvärderas som bättre än en tidtabell med långa körtider. Om en tidtabell är optimal innebär det att det **inte existerar** någon annan giltig tidtabell som har ett bättre målfunktionsvärde. Alla metoder som vi presenterar i denna rapport använder en målfunktion för att generera lösningar som är bra givet denna målfunktion, men alla metoder returnerar inte en optimal lösning. Metoder som är utarbetade för att returnera bra, men inte nödvändigtvis optimala, lösningar kallas *heuristiker*. Heuristiker används ofta när det tar för lång tid hitta den optimala lösningen. Tidtabellproblemet är ett exempel på ett NP-svårt problem, och exekveringstiden kan bli ohållbart lång.

Den första ansatsen för automatisk tidtabellläggning som utvecklades i FLTP är att använda en optimeringsmodell och rullande planering. Avsnitt 5.1 presenterar optimeringsmodellen som implementerades i FLTP, och rullande planering introduceras i avsnitt 5.2. Det var denna ansats som användes i fallstudien Skymossen-Mjölby, vars resultat presenterats ovan i avsnitt 4.2. Eftersom tidtabellproblemet blir för stort för optimeringsmodellen om större geografiska områden ska hanteras så undersöktes även olika sätt att använda modellen i heuristiska ramverk. Dessa heuristiker presenteras i avsnitt 5.3. I avsnitt 5.4 presenteras resultaten av provkörningar med de olika heuristikerna och optimeringsmodellen på Hallsbergs driftledningsområde.

5.1 Matematisk optimeringsmodell

Optimeringsmodellen består av de villkor som en giltig tidtabell måste uppfylla, samt en målfunktion som kan användas för att avgöra hur pass bra olika lösningar, dvs. tidtabeller, är. Givet denna optimeringsmodell kan sedan algoritmer användas för leta igenom lösningsrymden och hitta den optimala lösningen.

Ett vanligt koncept som används i optimering är *optimalitetsgap*. Optimeringsalgoritmer kan ge en gräns för hur pass bra målfunktionsvärdet kan bli, och givet denna gräns kan man räkna ut hur pass långt från optimalitet en given lösning max kan vara. Den modell som används i FLTP är en minimeringsmodell som minimerar körtid eller avstånd från önskade avtalstider. Därför är gränsen

på målfunktionen en undre gräns som visar hur pass litet målfunktionsvärdet kan bli. Säg t.ex. att vi ska minimera summan av alla tågs körtid. En trivial undre gräns är då summan av den körtid som tågen har om konflikter tillåts och tågen kör så snabbt som det tekniskt bara går från avgångs- till ankomststation. Summan av alla tågs körtid i en konfliktfri tidtabell kommer vara större eller lika med denna triviala undre gräns. Allt eftersom optimeringsalgoritmen letar igenom sökrymden med konfliktfria tidtabeller så kan den undre gränsen uppdateras, och samtidigt kan bättre och bättre giltiga tidtabeller hittas. När en lösning har hittats som har samma målfunktionsvärde som den undre gränsen vet vi att lösningen är optimal.

Optimeringsmodellen i FLTP baserades på en tidigare modell från [7] och [6]. Modellen i [7] och [6] är framtagen för att kunna användas i operativ drift, och bör således vara snabb. Vidare har den en nollpunkt vilket gör den lämplig för rullande planering. Modellen från [7] och [6] utvecklades sedan i FLTP för att passa det svenska tågssystemet, och idéer från den tidigare SICS-modellen Marackasen [2] inkorporerades. Till exempel så använder vi domäner som beskriver hur pass lång från det ansökta tågläget som ett tåg får ligga. För en mer omfattande och matematisk beskrivning av modellen hänvisar vi till [4], och i resten av det här avsnittet förklarar vi istället i stora drag vad som är implementerat i modellen.

5.1.1 Data i modellen

Optimeringsmodellens data är samma som TrainPlans. Detta innebär att stationer är modellerade som punkter med en viss kapacitet, och spår mellan stationer som länkar. Data för det geografiska järnvägsnätet och tåg exporterades från TrainPlan för den fastställda versionen av T14.

5.1.2 Tågs basekvationer

Optimeringsmodellen ser till att ett tåg går samma väg som specificerats i den fastställda versionen av T14, samt att det stannar på alla platser med kommersiell aktivitet. Faktum är att optimeringsmodellen tvingar alla tåg att stå still på alla platser med kommersiell aktivitet så pass länge som den fastställda tågplanen specificerar. Så om ett passagerartåg P stannar vid Hallsbergs personbangård för att släppa av passagerare, och det stoppet är 4 minuter långt, så kommer optimeringsmodellen se till att tåget alltid stannar minst 4 minuter vid Hallsbergs personbangård. Modellen kommer således inte själv bestämma minsta stilleståndstid vid stationer, och inte heller göra egna nodtillägg, även om detta kan implementeras om tydliga regler finns. Modellen innehåller inte heller några tillägg för sidotågvägar eftersom spår på stationer inte ingår i modellen.

Tåg kan behöva stanna även utan att det finns någon kommersiell aktivitet. T.ex. behöver tåg ibland stanna för att bli omkörda. Alla tåg kan inte stanna på alla geografiska punkter. Eftersom vi inte vet vilka tåg som kan stanna på vilka platser tillät optimeringsmodellen bara att tåg stannade på platser där de stannat i den fastställda tågplanen. Detta är en förenkling i våra exempel, och mer rättvisande indata vad gäller vilka tåg som kan stanna var bör användas i en verklig implementation.

För att avgöra hur snabbt ett tåg kan köra på en länk använder optimeringsmodellen gångtidsmallarna från TrainPlan. Tågets minsta traverseringstid

på en länk beror på huruvida tåget stått still eller inte i början och slutet av länken. Låt S betyda att tåget står still, och F att tåget är i rörelse. Då finns det fyra gångtidsmallsmönster: S-S, F-S, S-F och F-F. Den rätta gångtidsmallen väljs av optimeringsmodellen baserat på om tåget står still eller inte.

5.1.3 Konfliktreglering

Tåg i samma riktning

För tåg som kör i samma riktning kommer optimeringsmodellen på enkelspårsträckor att använda reglerna för “Linje med linjeblockering” från TH601 [1] och annars används en headway-tid på 3 minuter vid avgång och ankomst till tidtabells-punkter. Alla geografipunkter anses ha blocksignaler, men detta kan ändras om data för vilka punkter som har signaler finns.

Tåg i olika riktning

Tåg som kör i olika riktning får möta varandra på stationer på enkelspår, och då används de regler för “Tidsintervall vid ankomst” som finns i TH601 [1]. Det tre olika stationsmodellerna (ingen samtidighet, skyddsväxlar/avstånd och samtidig infart) ger upphov till olika regler i optimeringsmodellen. Det första tåget som ankommer till stationen för omkörning tvingas alltid stanna, dvs modellen tillåter inte flygande möten. Ingen tidsutsträckning krävs mellan avgångarna.

Tåg i olika riktning kan mötas på dubbelspårslinjer om inget av tågen kör på högerspår. Tågen får alltid mötas på dubbelspårsstationer utan tidspåslag så länge tågen kör på olika spår på båda sidor av stationen. Om tågen kör på samma spår på någon sida av stationen (t.ex. för att ett av tågen kör på högerspår) så används samma regler som för enkelspårsstation.

Stationer

Varje station har en maxgräns för hur många tåg som får befinna sig på stationen samtidigt. Denna maxgräns respekteras av modellen. Till exempel får det max finnas 3 tåg samtidigt i Töreboda, och då finns regler i optimeringsmodellen som förbjuder att fler än 3 tåg befinner sig i Töreboda samtidigt. Stationsspår finns inte med i modellen, utan de representeras bara av denna maxgräns.

5.1.4 Målfunktion

Målfunktionen är den matematiska funktion som används för att avgöra vilken av två tidtabeller som är bäst. Målfunktionen är väldigt viktig eftersom det är den som styr optimeringen och vilken lösning som väljs. Olika målfunktioner kan också ge olika exekveringstider.

Under FLTP har olika målfunktioner används, men oftast har summan av tågens körtid eller summan av feltider minimerats. Med feltid menas hur pass mycket senare ett tåg ankommer en avtalspunkt eller hur pass mycket tidigare ett tåg avgår från en avtalspunkt, jämfört med den önskade tiden (se tidigare Figur 9). Olika tåglägesaspekter för olika tåglägen kan läggas in i målfunktionen, men det är viktigt att de olika aspekterna viktas mot varandra på rätt sätt. I fallet med körtid vs. feltid kanske 15 minuters ökning i körtid för ett tåg anses vara mer problematiskt än 3 minuters feltid på 5 avtalspunkter för ett annat, men om målfunktionen är summan av körtider och feltider så kommer dessa två utfall få samma målfunktionsvärde och optimeringen kommer hantera

tidtabellerna som likvärdiga. Likaså kan det vara värt att lägga på en extra kostnad när ett mätvärde (dvs. en körtid eller feltid) försämras väldigt mycket. Till exempel så är det kanske värre att ett tåg får en 45 minuters ökning i körtid än att tre tåg får en 15 minuters ökning i körtid. Att använda en målfunktion som straffar stora försämringar för enstaka mätvärden mer än små försämringar för många mätvärden är typiskt något som ofta ger längre exekveringstider.

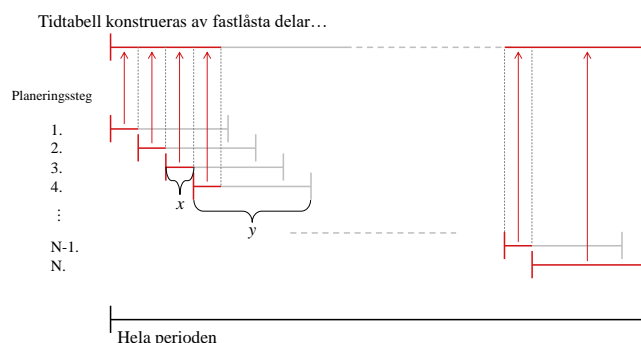
I fallstudien på banan mellan Skymossen och Mjölby minimerades två olika tåglägesaspekter beroende på operatörernas simulerade önskningar. Om operatören önskade en kort körtid så lades tågets totala körtid in i målfunktionen, och annars summan av feltider på tågets alla avtalspunkter. Ingen viktning av de olika aspekterna användes, och målfunktionen straffade inte enstaka stora försämringar mer än flera små.

5.2 Rullande planering

Bild 12 visar hur rullande planering går till. Metoden går ut på att en plan genereras för en lång tidsperiod genom att man planera för kortare, överlappande, tidsperioder och sedan pusslar ihop dellösningarna. Oftast börjar man med den planeringsperiod som ligger tidigast och jobbar sig sedan framåt i tiden. Alla senare planeringsperioder måste då respektera de beslut som tagits i tidigare perioder och som nu inte går att ändra längre.

I FLTP var det en tidtabell för tidsperioden 1 januari till 13 december 2014 som genererades. En planeringsperiod på 24 timmar användes, och 12 timmar användes som steglängd mellan de olika planeringsperioderna. Detta innebär att den första uppgiften är att optimera fram en tidtabell för perioden 00:00 01-01-2014 till 00:00 02-01-2014, och sedan låsa fast alla tåglägen mellan 00:00 och 12:00 01-01-2014. Efter det genereras nästa probleminstans som sträcker sig mellan 12:00 01-01-2014 och 12:00 02-01-2014 och som måste respektera alla fastlåsningar som gjorts i perioden 00:00 till 12:00 01-01-2014. Detta innebär att den andra probleminstansen måste respektera var i geografin som tågen befinner sig vid 12:00, samt om de stannat på stationen innan eller inte (för att veta vilken gångtidsmall som ska användas). Genom att senare instanser alltid respekterar de fastlåsta delarna av alla tåglägen så garanteras att de fastlåsta tåglägesdelarna från varje instans kan kombineras ihop till en tidtabell för hela den ursprungliga tidsperioden. Den rullande planeringen fortsätter enligt mönstret fastlåsning, probleminstansgenerering och optimering tills sista instansen är löst, 00:00 13-12-2014 till 00:00 14-12-2014. Sedan sätts alla fastlåsta delar ihop till en tidtabell för hela året.

Rullande planering garanterar inte att årstidtabellen är optimal. Genom att planera med långa tidsperioder och använda stora överlapp mellan instanserna så ökar chansen att tidtabellen är optimal eller nästan optimal. Men det finns inga garantier. Det kan till och med vara så att om planeringsperiodslängden eller steglängden inte är väl avvägda, eller om problemet är olämpligt för rullande planering, så kan tidigare fastlåsningar leda till att en senare probleminstans blir olösbar. Men detta var inte något som vi hade problem med i de test som gjordes i FLTP.



Figur 12: Vid rullande planering delas tidsperioden upp i bitar av storlek y som börjar var x :e tidsperiod. Varje tidtabellsbit optimeras i tidsordning, och den del av tidtabellen som ligger innan nästa tidtabellsbits början, blir fastlåst (röda bitar). Nästa tidtabellsbit måste respektera att fastlåsta tåg är på en viss geografi vid tidtabellsbitens början. Alla fastlåsta bitar kan sedan läggas samman till en tidtabell för hela tidsperioden.

5.3 Heuristiska metoder

Även om optimeringsmodellen som beskrivs i 5.1 gick bra att använda i testfallet på banan mellan Skymossen och Mjölby så blir exekveringstiderna för långa om en tidtabell ska optimeras fram för ett större geografiskt område. Därför tog FLTP fram heuristiska ramverk som kan användas tillsammans med optimeringsmodellen för att generera tidtabeller för större geografier. Den första heuristiken (avsnitt 5.3.1) lägger till tåg i omgångar och bestämmer hur de ska köra, medan den andra (avsnitt 5.3.2) löser tidtabellproblemet för mindre geografibitar och försöker sedan pussla ihop tidtabellerna i kanterna där områdena möts. Sist, i avsnitt 5.3.3, beskrivs också en förbättringsheuristik som försöker förbättra en given, suboptimal, tidtabell. Heuristikerna testades på Hallsbergs driftledningsområde och testresultaten presenteras i avsnitt 5.4.

5.3.1 Successiv tilläggs- och fixerings-heuristik (STFH)

Den successiva tilläggs- och fixerings-heuristiken, STFH, försöker generera en tidtabell genom att lägga in tåg i tidtabellen i omgångar och fixera mötesplatser allt eftersom. Låt $b \in \mathcal{B}$ vara de batcher av tåg som ska läggas in i tidtabellen, och låt det nedsänkta indexet representera ordningen som batcherna läggs till. Dvs. låt b_1 vara den första tågbatchen som läggs in i tidtabellen, och b_2 den andra, och så vidare.

Den allra första batchen av tåg, b_1 , läggs in i en helt tom tidtabell och konfliktregleras. Alla mötesplatser fixeras och resultatet är en tidtabell med tågen i b_1 . Efter detta läggs nästa tågbatch b_2 in. Dessa tåg konfliktregleras med både de mötesfixerade tågen (b_1) och med sig själva (b_2), och alla mötesplatser fixeras. Tidtabellen består nu av alla tåg i b_1 och b_2 och dessa är mötesfixerade. Efter det läggs nästa batch, b_3 , till, och så vidare.

Det finns en risk att när en batch b_n läggs till blir tidtabellproblemet omöjligt att lösa på grund av de tidigare mötesfixerade tågen (se Figur 13). För att försöka återgå till ett lösbart problem så släpps mötesplatsfixeringen för alla

tåg som kan hamna i konflikt med något av tågen i batch b_n . Låt $K(i)$ vara alla tåg som ett tåg i kan hamna i konflikt med. Då ges den mängd tågs vars mötesplatsfixeringar vi ska släppa av $\mathcal{I}^1 = \{K(i), i \in b_n\}$ och vi kallar det nya problemet för det *1-relaxerade problemet*. En ny optimering startas på det 1-relaxerade problemet. Om det 1-relaxerade problemet är lösbart kommer en giltig tidtabell hittas, och alla mötesplatser i den nya tidtabellen kan fixeras och algoritmen gå vidare till att lägga till alla tåg i nästa batch, b_{n+1} . Om däremot det 1-relaxerade problemet också är omöjligt att lösa så fortsätter relaxeringen genom att mötesplatsfixeringen släpps för alla tåg som har möte med något av tågen i \mathcal{I}^1 eller b_n , $\mathcal{I}^1 \cup b_n$. Detta blir det 2-relaxerade problemet och tågen vars mötesplatsfixering ska släppas är $\mathcal{I}^2 = \{K(i), i \in \mathcal{I}^1 \cup b_n\}$. Denna släppning av mötesplatsfixering (relaxering av problemet) fortsätter tills problemet antingen blir lösbart och algoritmen kan återgå till att lägga till nya tågbatcher, eller tills det inte finns fler tåg vars mötesplatsfixering vi kan släppa. Om alla mötande tåg är släppta och problemet fortfarande är olösbart så innebär det att det ursprungliga tidtabellproblemet är olösbart.

En mer utförlig beskrivning av den successiva tilläggs- och fixerings-algoritmen, på engelska kallad *Incremental addition and fixing heuristics*, finns i [5].

5.3.2 Iterativ geografisk pusslingsheuristik (IGPH)

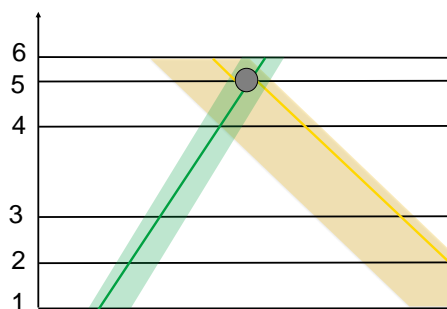
I det här avsnittet beskrivs en iterativa geografiska pusslingsheuristik (IGPH) som försöker generera en tidtabell genom att dela upp geografien i mindre, lösbara, geografi-bitar och sedan pussla ihop dessa bitar till en giltig tidtabell.

Låt oss kalla den totala geografien T och geografibitar $g \in G$, låt oss vidare kalla det totala problemet TP och delproblemen för de olika geografibitar $DP - g$. Tidtabellspunkter som ligger på gränsen mellan två geografibitar kallas *gränspunkter*, och två delproblem som delar gränspunkt(er) kallas grannproblem. Målet med algoritmen är att hitta tidtabeller för alla delproblem som matchar på alla gränspunkter. Figur 14 visar den bakomliggande tanken med IGPH.

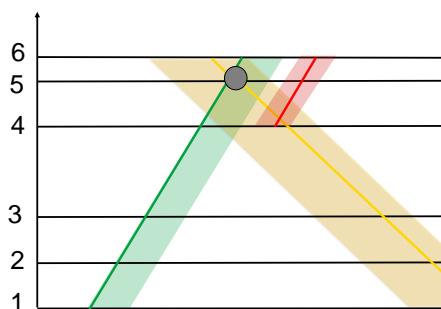
Algoritmen har två stadier. Det första stadiet är initiering, och består av att alla delproblemen för $g \in G$ löses separat. Detta ger dellösningar, eller deltidtabeller. Algoritmen övergår sedan till det andra stadiet, nämligen en pusslingsfas som itererar mellan två strategier tills en giltig tidtabell hittats eller en tidsgräns för nedstängning nåtts. De två pusslingsstrategierna är:

1. Lös en reducerad och relaxerad variant av det totala problemet, kallat $RR - TP$, med mål att i första hand minska grannlösningars oförenlighet och i andra hand ge bra tåglägen.
2. Optimera fram nya tidtabeller för alla delproblemen $DP - g, g \in G$ med mål att först och främst passa ihop med grannlösningar och i andra hand ge bra tåglägen.

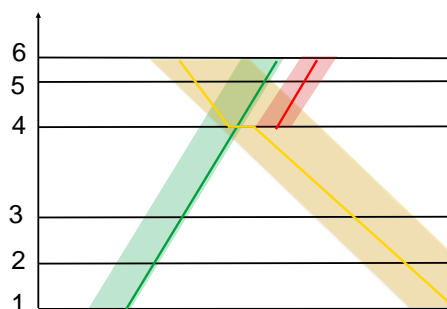
De två strategierna beskrivs utförligare nedan. Notera att det inte finns två nackdelar med metoden; det finns inte någon garanti för att IGPH avslutas, och det finns inte något sätt att avgöra om ett tidtabellproblem är lösbart eller inte.



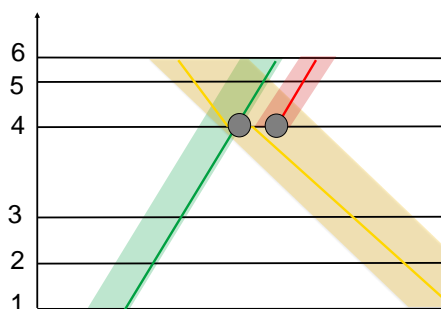
(a) Det gröna och det gula tåget läggs in i den första batchen och mötesplatsen fixeras.



(b) Det röda tåget ska nu läggas in i tidtabellen, men det går inte att konstruera en konfliktfri tidtabell om det gröna och det gula tåget ska mötas på station 6. Tidtabellproblemet är omöjligt att lösa.

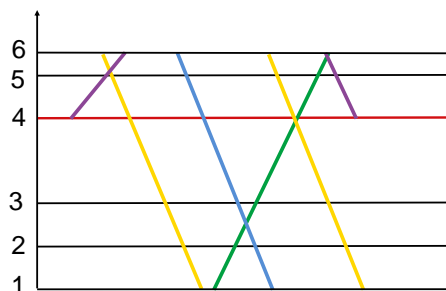


(c) I ett försök att göra tidtabellproblemet lösbart tas mötesplatsfixeringen bort för alla tåg som kan vara i konflikt med det röda tåget, och sedan löser vi det 1-relaxerade tidtabellproblemet.

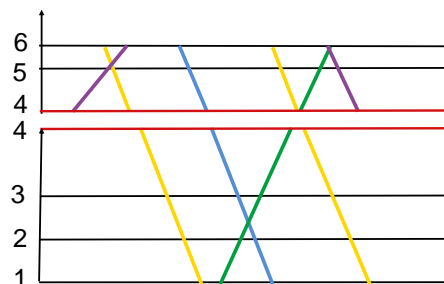


(d) Det nya tidtabellproblemet är lösbart och en tidtabell hittas. Alla mötesplatser fixeras, och algoritmen går sedan vidare till att lägga till nästa batch med tåg.

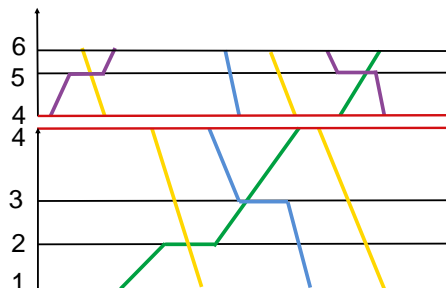
Figur 13: Ett grönt, ett gult och ett rött tåg ska köra på en enkelspårsbana. Alla tåg måste gå inom sina skuggade tidsdomäner. Det gula och gröna tåget läggs in i tidtabellen i batch ett, och sedan läggs det röda tåget in. Fixeringen av mötesplatsen för det gröna och gula tåget gör tidtabellproblemet olösbart när det röda tåget ska läggas in. Fixeringen måste släppas och det 1-relaxerade tidtabellproblemet löses.



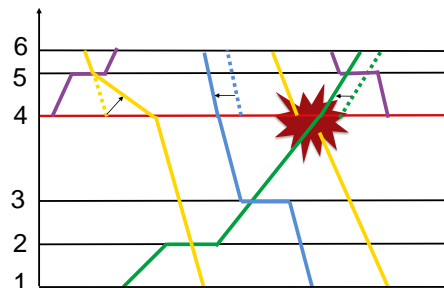
(a) Hela problemet TP ska delas upp i två. Den rödmarkerade stationen är gränspunkten.



(b) Det totala problemet har delats upp i två delproblem.



(c) Delproblemen är tillräckligt små för att de ska gå att optimera fram en tidtabell för dem. Vi får då dellösningar.



(d) För att få en giltig tidtabell för det totala problemet måste dellösningarna ha tåglägen som passar ihop vid gränspunkterna. I exemplet uppstår en olöst konflikt mellan det gröna och det gula tåget eftersom de inte har möts och konfliktreglerats i någon av dellösningarna, och inget av tågen får stanna på gränspunkten, så mötet kan inte genomföras där.

Figur 14: Exempel på hur en total geografi T delas upp i två mindre $g \in G$. Delproblemen i geografibitarna är tillräckligt små för att en optimeringsmotor ska kunna hitta en tidtabell.

Initiering: Lösa alla delproblem separat

Det första steget i IGPH är att hitta en giltig tidtabell för alla delproblem. Målfunktionen som används i IGPH är att minska avståndet från de ansökta avtalstiderna, dvs. minimera feltiden. Genom att minimera summan av feltider så kommer tåglägena generellt passa ganska bra ihop efter initieringsfasen. Om däremot målfunktionen skulle vara att t.ex. minimera summan av tågens körtid så kan tåglägena vara helt utspridda, vilket ger en mycket svårare pusslingsfas. Resultatet av initieringen är att det finns deltidtabeller för alla geografibitar $g \in G$.

Pusselstrategi 1: Reducerat och relaxerat totalproblem ($RR-TP$)

Pusselstrategi 1 baserar sig på att totalproblemet reduceras så pass mycket att hela geografiområdet kan hanteras i optimeringsmodellen. Reduceringen består av att alla mötesplatser och stoppmönster fixeras i enlighet med dellösningarna från alla problem $DP - g, g \in G$. Så om tåg A och tåg B möts i Laxå som ligger mitt i en geografibit, så kommer mötet att fixeras till Laxå i totalproblemet. Dessvärre kan fixeringen av mötesplatser och stopp leda till att det totala tidtabellproblemet blir olösbart. Figur 15a till 15c visar tre olika exempel på olösbara problem. För att komma runt det reducerade problemets olöslighet relaxeras totalproblemet vid alla gränspunkter, dvs. relevanta villkor mjukas upp och får brytas, men varje brott straffas med en så pass hög kostnad i målfunktionen att denna kostnad blir dominant. Målfunktionen i optimeringen av det reducerade och relaxerade totalproblemet blir nu först och främst att hitta en giltig tidtabell, och i andra hand att minimera summan av feltid.

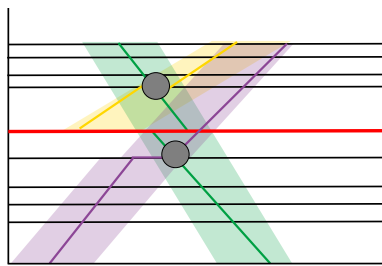
Ofta kommer det inte vara möjligt att hitta en giltig tidtabell, utan de relaxeringar som gjorts kommer användas trots att de är straffade med en hög kostnad i målfunktionen. Algoritmen kommer då släppa fixeringen av mötesplatser och stopp för de tåg som inte är giltiga (se Figur 15d). Om alla ogiltiga tåg släpps samtidigt kan problemet blir för stort för att kunna hanteras i optimeringsmodellen. Därför delas de ogiltiga tågen upp i mindre grupper som släpps en efter en. När en grupp ogiltiga tåg släppts används optimeringsmotorn för att försöka hitta en mer eller helt giltig tidtabell. När en lösning returnerats av optimeringsmotorn så fixeras de släppta tågen, och, om det finns fler ogiltiga tåg som ännu inte släppts, så släpps nästa grupp ogiltiga tåg.

Om en giltig tidtabell hittas efter att det relaxerade och reducerade totalproblemet lösts, eller efter att fixeringen av ogiltiga tåg släppts och optimeringen gjorts om, så returneras den giltiga tidtabellen. Annars går algoritmen vidare till pusselstrategi 2.

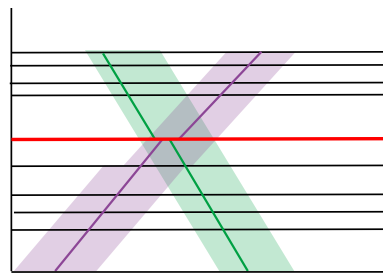
Pusselstrategi 2: Om-optimering av delproblem

Målet med pusselstrategi 2 är att lösa om alla delproblem och få antingen dellösningar som passar ihop eller nya mötesplats- och stopp-fixeringar att skicka in i pusselstrategi 1.

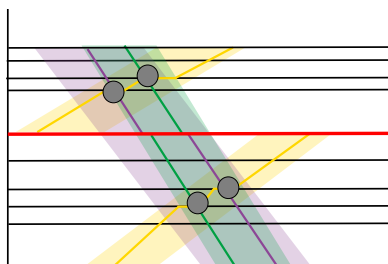
I initieringsfasen så optimeras tidtabeller för delproblemen fram helt separat och det är summan av feltider som minimeras. I pusselstrategi 2 försöker optimeringen i första hand träffa måltidpunkter för alla tåglägen på gränspunkter, och i andra hand minimera summan av feltider. Alla måltidpunkter har en egen speciell kostnad i målfunktionen. Hur pass stor kostnaden är beror på hur pass många gånger tåget har varit ogiltigt, eller haft ett ogiltigt möte, i $RR - TP$. Målpunkterna för tåg beror också på om tåget eller något av dess möten var ogiltiga i $RR - TP$. Om tåget eller något av dess möten var ogiltiga kommer den gränstidpunkt som sattes av den sista optimeringen av $RR - TP$ att användas som måltidpunkt. Annars används den tid som grannlösningen har på gränspunkten.



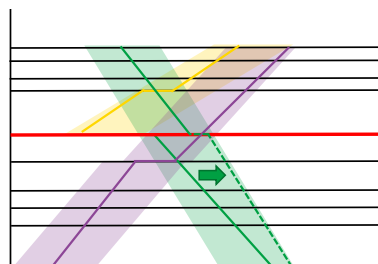
(a) De fixerade mötesplatserna gör så att det gröna tåget aldrig kan hålla ihop tidsmässigt. Tidsvillkoren relaxeras på gränspunkter.



(b) Båda delproblemen har satt en mötesplats utanför det egna geografiska området. Om tågen inte kan mötas på gränspunkten så är problemet olösbart. Konfliktvillkoren relaxeras på gränspunkten.



(c) De spikade fixerade mötesplatserna leder till att det gröna och lila tåget har olika tågordningar. Tågordningsvillkoret relaxeras på gränspunkter.



(d) Genom att släppa fixeringen för ogiltiga tåg kan en giltig tidtabell hittas. Här har fixeringarna för situationen i 15a släppts, och en giltig lösning kan hittas genom att det gröna tåget möter det lila tåget på gränspunktsstationen istället för på stationen efter gränspunkten.

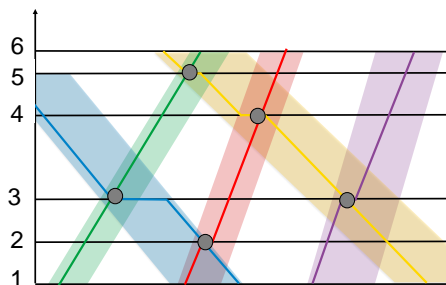
Figur 15: Exempel på hur ett problem kan bli olösbart när mötesplatser fixeras baserat på dellösningar. Den röda stationen är gränspunkten.

5.3.3 Förbättringsheuristik (FH)

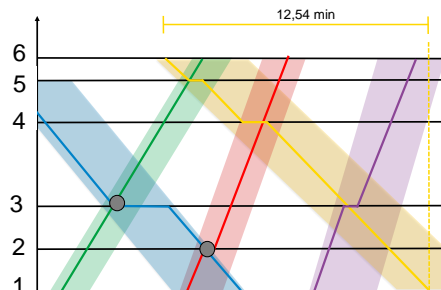
En tidtabell som genererats med någon av de heuristiska metoderna ovan behöver inte alls vara optimal. En enkel förbättringsheuristik togs därför fram för att förbättra kvaliteten på de heuristiskt genererade tidtabellerna.

Förbättringsheuristiken börjar med att alla mötesplatser och stoppmönster från den ursprungliga tidtabellen fixeras. Det innebär att tidtabellproblemet är reducerat och går snabbt att lösa för optimeringsmotorn. Sedan släpps mötes- och stopp-fixeringen för ett antal tåg, och optimeringsmotorn tillåts lägga in de frisläppta tågen där de passar bäst in givet den specificerade målfunktionen. Detta upprepas tills alla tåg har släppts och om-optimerats en gång. Figur 16 är ett exempel på hur förbättringsheuristiken fungerar i en liten tidtabell med

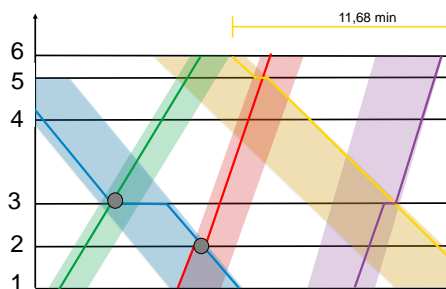
fyra tåg och där målfunktionen är att minimera summan av körtider.



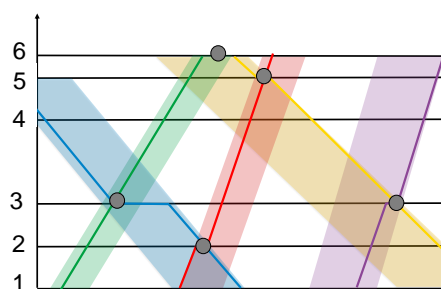
(a) Alla mötesplatser och stoppmönster fixeras i den giltiga tidtabeller som ska förbättras.



(b) Fixeringen för det gula tåget släpps. Med hjälp av optimering kan det gula tåget läggas in i tidtabellen igen på det sätt som minimerar målfunktionen. Det gula tågets körtid innan om-optimering är 12,54 minuter och visas som ett streck.



(c) Det gula tåget har flyttats så att det har fått en kortare körtid på 11,58 minuter. Alla andra tåg har samma körtid som innan och tidtabellens målfunktionen har således blivit bättre.

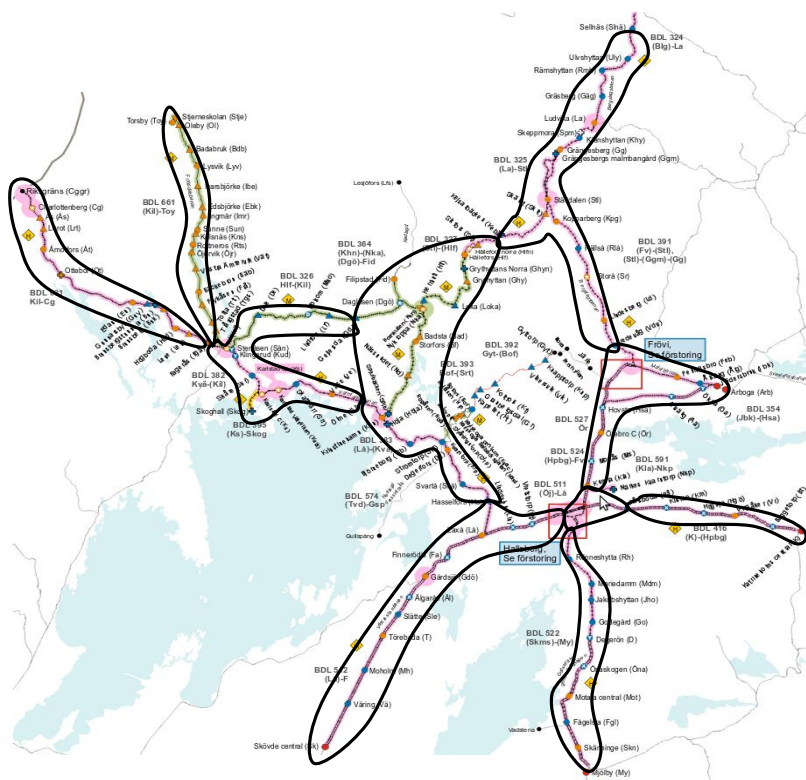


(d) Det gula tåget fixeras igen, och ett nytt tåg kan släppas.

Figur 16: Exempel på hur förbättringsheuristiken fungerar i en liten tidtabell med fyra tåg. Målet är att minimera summan av alla körtider.

5.4 Testfall: Hallsbergs driftledningsområde

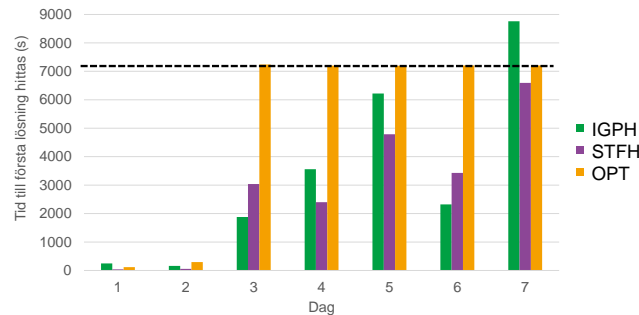
STFH och IGPH testades på Hallsbergs driftledningsområde, och jämfördes med optimeringsmotorn i den kommersiella optimeringsprogramvaran CPLEX 12.2. En tidtabell genererades för varje dag i första veckan i februari 2014. De två första dagarna är helgdagar, och då går ca 300 tåg. De andra dagarna är veckodagar, och då går ca 600 tåg. Figur 18 visar hur lång tid det tog innan en första giltig tidtabell returnerades av de olika metoderna. När vanlig optimering i CPLEX 12.2 användes så stängdes optimeringen ner efter 2 timmar, medan heuristikerna fick fortsätta tills en giltig lösning genererats. Målfunktionen för standard CPLEX och STFH var att minimera summan av alla körtider, medan IGPH minimerade avstånd från ansökningar.



Figur 17: Hallsbergs driftledningsområde och den uppdelning som användes i IGPH.

STFH var den enda metod som alltid returnerade en giltig tidtabell inom 2 timmar, och den var snabbast för 5 av de 7 dagarna. De andra två dagarna var IGPH snabbast. För de större problemen med uppemot 600 tåg returnerade standard CPLEX optimering aldrig någon lösning inom 2 timmar.

När det gäller förbättringsheuristiken, FH, jämfördes den med två funktioner i CPLEX. Den jämfördes dels med optimeringsfunktionen och dels med en inbyggd förbättringsheuristik, kallad Polishing. Alla tre metoder startades från



Figur 18: Jämförelse av de olika lösningsmetoderna. Den normala CPLEX optimeringen stängdes ner efter två timmar (streckade linjen). STFH hittar alltid en lösning inom två timmar, och är den snabbaste metoden för fem av sju dagar. De andra två dagarna är IGPH snabbast.

den lösning som STFH genererat. FH får köra tills den slutar, dvs. tills alla tåg har tagits bort och lagts till en gång, medan CPLEX standard optimering och Polishing tillåts lösa i 10 minuter. I Figur 19 visas tidtabellens förbättring för de olika metoderna. Målfunktionsvärdet syns på y-axeln, och exekveringstiden på x-axeln, och varje gång som en ny tidtabell returneras markeras detta med en punkt.

För helgdagarna med lite trafik är CPLEX standard optimering och CPLEX Polishing bäst. FH är lika snabb som standardfunktionerna i CPLEX, men eftersom FH avslutas efter ca en minut när den har om-optimerat alla tågen en gång medan CPLEX standardfunktioner fortsätter i 10 minuter så hittar CPLEX standardfunktioner bättre lösningar. När det kommer till de större problemen för veckodagar så är däremot FH bättre. Den hittar bättre lösningar snabbare för fyra av fem dagar. Den femte dagen hittar FH lika bra lösningar som standardfunktionerna i CPLEX efter 10 minuter, men den är långsammare.

6 Sammanfattning

Målet med Framtidens LeveransTågplaneProcess projektet var att undersöka planeringsmetoder för att ta fram leveransåtagande i långtidsprocessen då Successiv Planering är fullt genomfört. I början av projektet identifierades fyra mål för planeringsmetoden, den skulle (1) vara snabb, (2) ta fram leveransåtaganden som uppfyller olika behov för olika sökanden, (3) ta fram leveransåtaganden som ger ett lösbart tidtabellproblem varje driftdag och (4) ta fram leveransåtaganden som använder spårkapaciteten effektivt.

Eftersom Successiv Planering tillåter att tåg kör olika olika dagar så tog vi i FLTP fram en planeringsmetod för långtidsprocessen som beaktar kördagsvariationer. Mer precist så genereras kontrolltidtabeller för olika tidsperioder, och tåglägena från dessa tidtabeller analyseras för att bestämma vilka leveransåtaganden som bör erbjudas de sökande. En analys av uteblivna konflikter i T14 pekade på att metoden skulle kunna leda till bättre leveransåtagande för vissa tåg, och ett försök på sträckan Skymossen-Mjölby stärkte denna hypotes ytterligare. Testet på sträckan Skymossen-Mjölby visade också att den föreslagna metoden kan

ta fram leveransåtagande som uppfyller olika behov för olika sökande. T.ex. så kan sökande antingen önska specifika avtalstider på platser med kommersiell verksamhet, eller en kort total körtid. Resultaten från testfallet visade att de leveransåtagande som genererades tydligt speglade de sökandes önskningar. Om en worst-case analys används för att bestämma leveransåtagande så garanterar metoden dessutom att en konfliktfri produktionstidtabell som uppfyller alla avtalstider alltid kan genereras. Att planera för varje dag individuellt i långtiden jämfördes med en optimal bastidtabell för sträckan Skymossen-Mjölby, och resultaten visade att daglig planering generellt ger bättre leveransåtagande vilket tyder på att den föreslagna metoden använder spårkapaciteten effektivt.

Av de mål som specificerades vid projektets början så är metodens snabbhet det mål som inte har uppfyllts helt. Eftersom den föreslagna planeringsmetoden baserar sig på att flera tidtabeller genereras så behövs en bra automatisk tidtabellsgenerator. Om stora geografiska områden ska hanteras tar det för lång tid att lösa optimeringsproblemet i CPLEX 12.2. Därför tog projektet fram heuristiska ramverk för optimeringsmodellen. Ett av dessa ramverk, successiv tilläggs- och fixerings-heuristik i kombination med en förbättringsheuristik, ger lovande resultat för Hallsbergs driftledningsområde. Det kvarstår dock arbete för att avgöra om metoden går att använda på hela Sverige, och ytterligare minskning av exekveringstid är önskvärt.

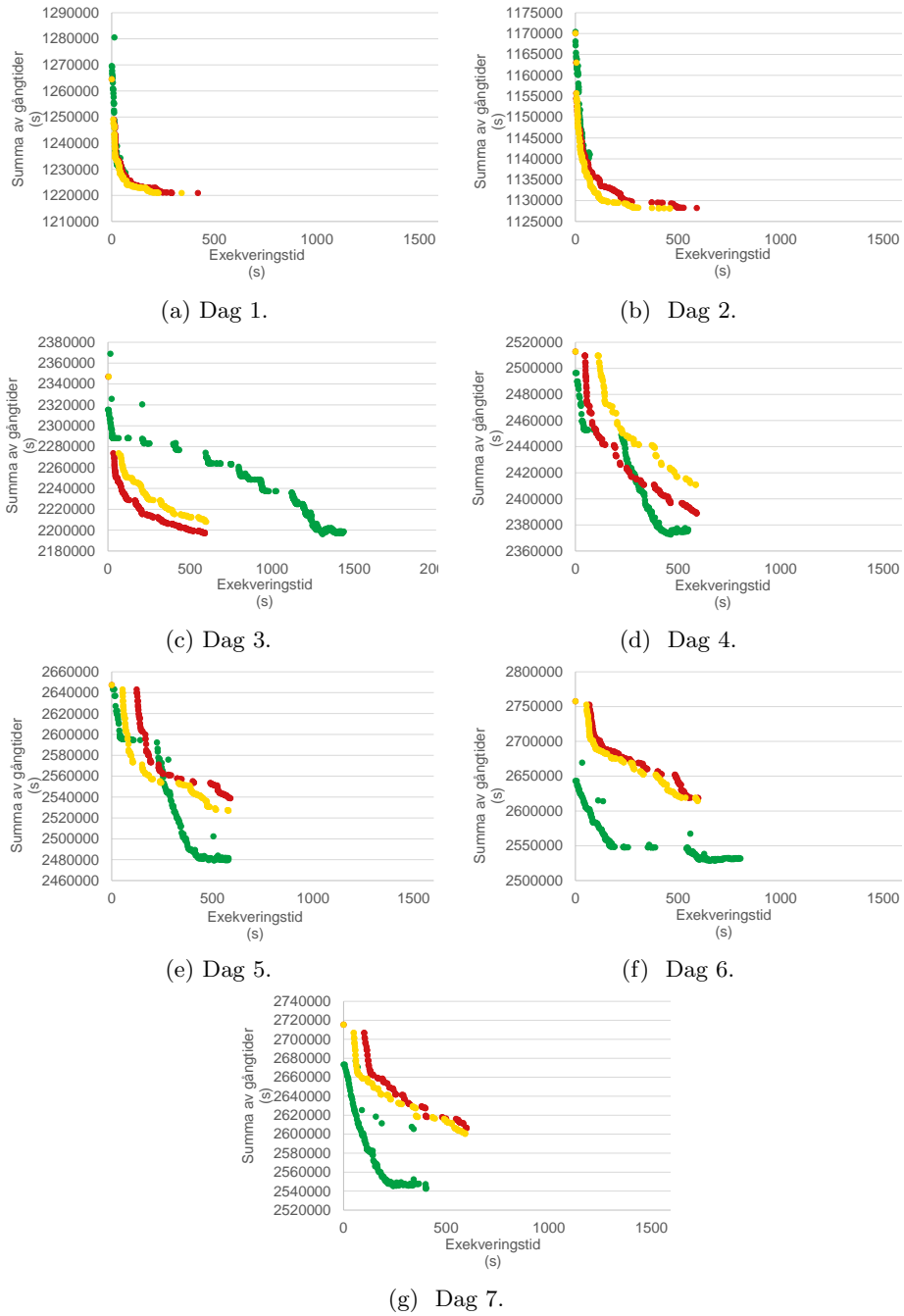
En annan frågeställning som kvarstår är hanteringen av problemet med att ett tåg ibland får ett fåtal tåglägen som är mycket sämre än de andra. En möjlig lösningsansats kan vara att målfunktionen viktas eller utformas annorlunda, eller att villkor på hur tåglägeskogen får se ut inkluderas i optimeringsmodellen. Vidare finns några metodspecifika aspekter som i dagsläget kan verka kontroversiella och som bör undersökas närmare, så som det faktum att en avtals-ankomsttid för ett tåg till en station ibland kan ligga efter tågets avtals-avgångstid från samma station.

Ett antal större frågeställningar som inte var fokus under projektets gång kvarstår också. Till exempel så har frågeställningen om hur man ska kunna spara kapacitet till ad hoc-tåg inte undersökts, och banarbetsplanering har inte ingått i projektet.

Referenser

- [1] Riktlinjer för tidtabellskonstruktion för tåg på statens spåranläggningar, TH 601. Technical report, Trafikverket, 2003.
- [2] M. Forsgren, M. Aronsson, and S. Gestrelus. Maintaining tracks and traffic flow at the same time. *Journal of Rail Transport Planning & Management*, 3(3):111–123, 2013.
- [3] S. Gestrelus. Advanced planning in the industry. http://tp2015.sics.se/Advanced_Planning_in_the_Industry.pdf, 2014.
- [4] S. Gestrelus, M. Aronsson, and M. Bohlin. On the uniqueness of operation days and delivery commitment generation for train timetables.
- [5] S. Gestrelus, M. Aronsson, and A. Peterson. A matheuristic for the train timetabling problem. In *Abstract accepted to Transportation Research Procedia (EWGT2017)*, 2017.

- [6] L. Lamorgese and C. Mannino. An exact decomposition approach for the real-time train dispatching problem. *Operations Research*, 63(1):48–64, 2015.
- [7] C. Manino. Real-time traffic control in railway systems. In *OASIs-OpenAccess Series in Informatics*, volume 20. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2011.



Figur 19: Jämförelse mellan förbättringsheuristik (grön), CPLEX standard (gul) och CPLEX Polishing (röd). Varje punkt visar när en tidtabell returneras. Förbättringsheuristiker genererar bättre tidtabeller snabbare än både standardkörning med CPLEX och CPLEX Polishing för de stora tidtabellproblemen Dag 4-7. Dag 3 är också ett stort tidtabellproblem, och då hittar alla tre metoderna lika bra lösningar.